



## 成都众山科技有限公司产品说明书

产品型号：YX-DID0-004

全部资料下载地址：<http://ask.zstel.com:8090>

技术支持服务电话：028-64267900

技术支持专员企业 QQ：3183329475

官网网站：<https://www.zstel.com/>

硬件/软件技术定制热线：19150158475 张工

# 目录

目录.....	2
一、 产品概述.....	3
1.1 概述.....	3
1.2 性能特点.....	3
1.3 功能汇总.....	3
1.4 技术参数.....	4
二、 外观尺寸.....	5
2.1 产品外观.....	5
2.2 产品尺寸图.....	5
三、 产品接线图、跳线、指示灯说明.....	8
3.1 接线图.....	8
3.2 LED 指示灯.....	9
四、 软件操作.....	10
4.1 配置软件.....	10
4.2 配置基本参数.....	11
4.3 DI 开关量相关参数.....	11
4.4 短信告警内容参数配置（设备需包含有短信模块）.....	13
五、 ModbusRTU 通讯协议、组态软件软件说明.....	15
5.1 通讯协议.....	15
5.2 寄存器地址.....	15
5.3 Modbus RTU 功能码.....	15
5.4 Modbus 通讯实例.....	15
六、 协议详解.....	17
6.1 功能码描述.....	17
6.2 错误码描述.....	24
6.3 CRC 校验算法.....	25
七、 附录.....	27

## 一、产品概述

### 1.1 概述

YX-DID0-004 是一款工业级标准开关量采集产品，共有 4 个开关量输入通道。每个通道均可以分别设置联动控制 DO 继电器输出；RS-485 通讯接口使用标准 Modbus RTU 协议，符合工业标准。

### 1.2 性能特点

- 防死机软、硬件看门狗
- 5~35V 带防反接、过压过流保护电源
- 4 路继电器带隔离常开，常闭输出
- 4 路带光耦隔离开关量输入
- 可配置高、低输入
- 高性能低功耗 32 位 ARM 嵌入式 CPU
- 支持 ModbusRTU 从站协议
- 多路指示灯
- 带防雷、静电保护 RS485 通讯接口
- 工业机温度范围，应对严苛现场环境

### 1.3 功能汇总

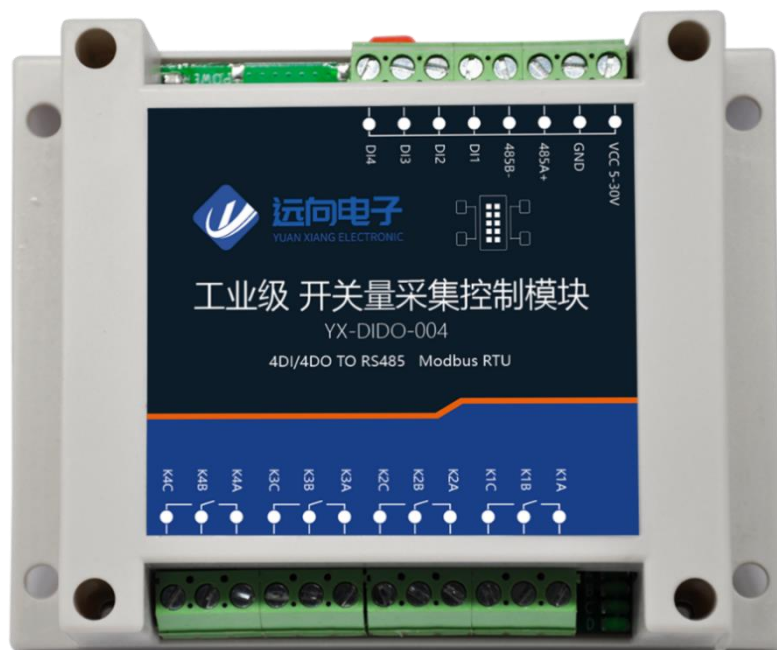
- 1) 4 路开关量输入
- 2) 4 路继电器常开、常闭输出
- 3) 支持 ModbusRTU 通讯协议
- 4) 自定义 1~255modbus 设备地址
- 5) 自定义短信告警内容（设备需包含有短信模块）
- 6) 输入信号振荡保持（可消除输入高/低信号不稳定跳动）
- 7) 灵活配置 DI-DO 联动控制
- 8) DI 支持可配置电平触发
- 9) 支持各类组态软件、触摸屏
- 10) 支持闪开闪断功能

1.4 技术参数

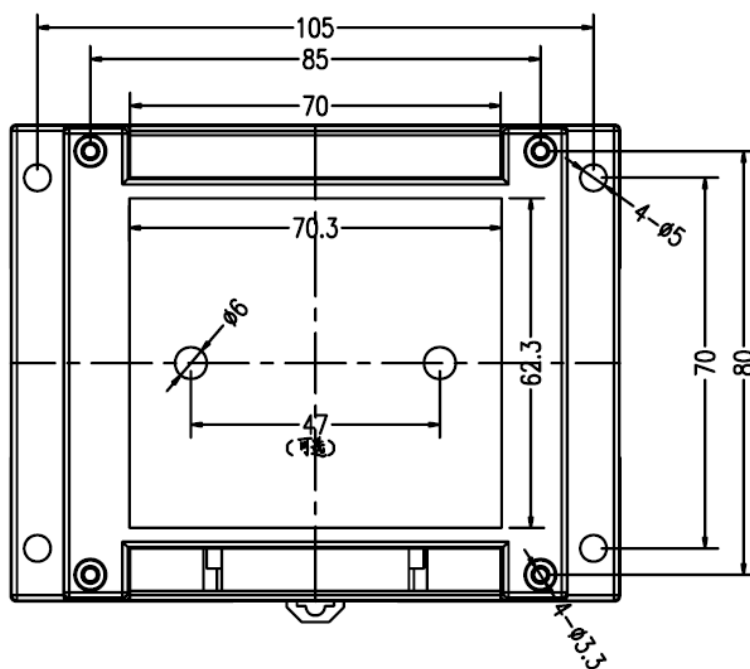
开关量输入 接口	DI	4 路单端
	DI 响应	下降沿，NPN（低）输入
	响应时间	30ms
	输入类型	干节点、无源（低）输入
	最大输入电压	电源输入电压
	最大输入电流	20mA
	输入阻抗	2.7kΩ
继电器输出	继电器类型	4 路继电器 常开、常闭
	触电电阻	100mΩ
	机械寿命	1x10 <sup>6</sup> 次
	最大切换电压	0~250VAC，0~30VDC
	最大切换电流	0~10A
通讯接口	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	N. 8. 1
	通讯协议	ModbusRTU
电源参数	电源规格	DC5~35V
	功耗	12V-500mA
工作环境	工作温度、湿度	-40℃~85℃，0%RH~95%RH
其他	尺寸	115*90*40

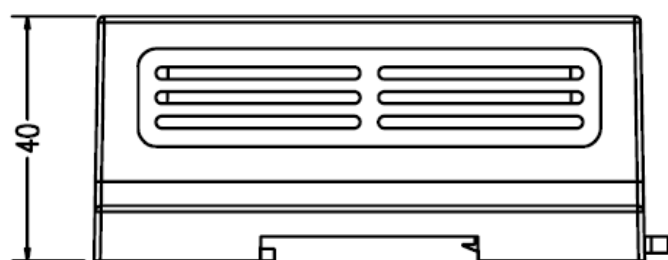
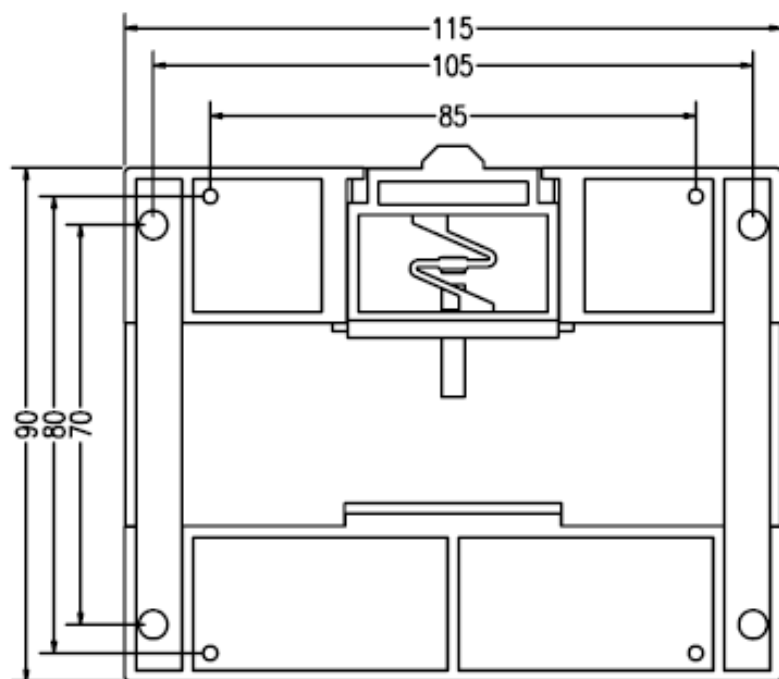
## 二、 外观尺寸

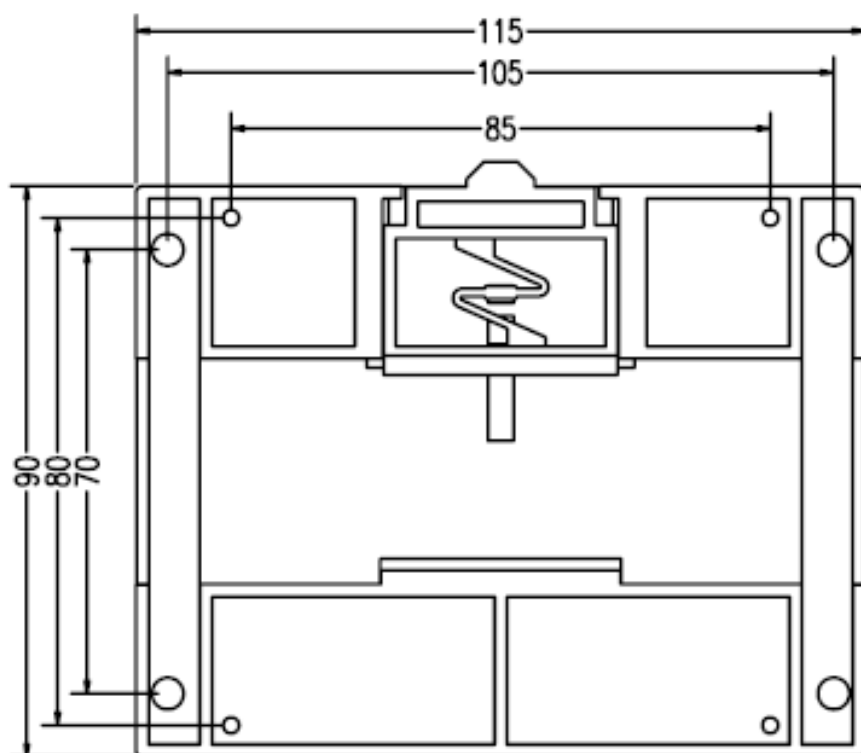
### 2.1 产品外观



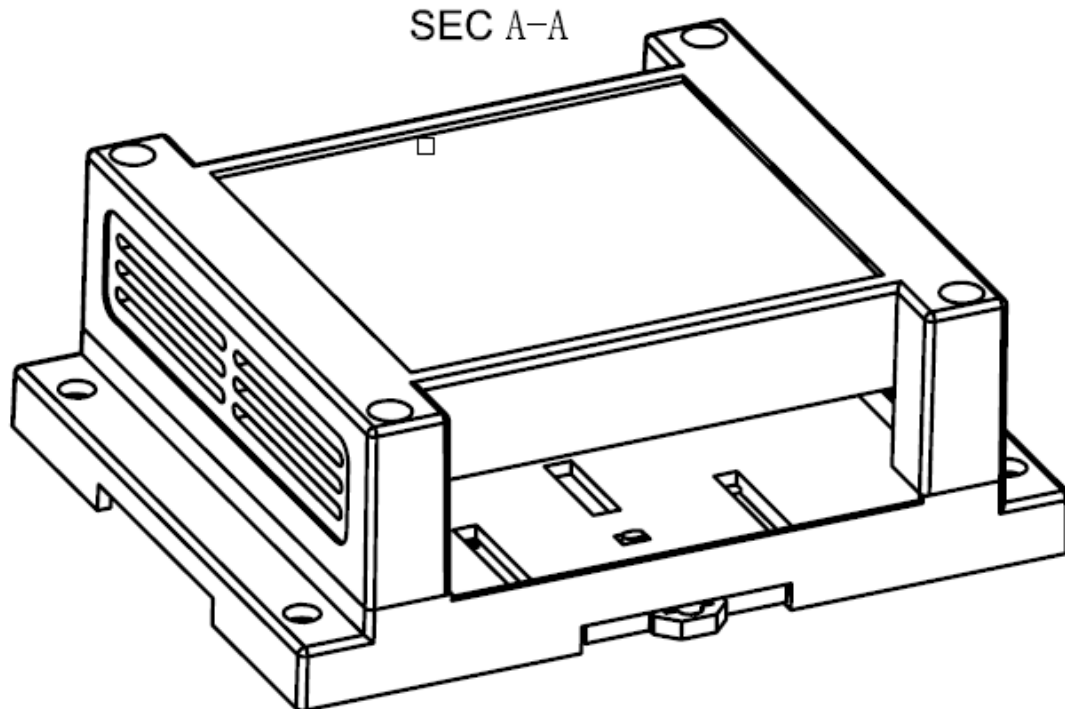
### 2.2 产品尺寸图







SEC A-A



### 三、 产品接线图、跳线、指示灯说明

#### 3.1 接线图



上端：

VCC： 电源正极  
GND： 电源负极  
A+： 485 A+  
B-： 485 B-

DI1： 开关量输入通道 1  
DI2： 开关量输入通道 2  
DI3： 开关量输入通道 3  
DI4： 开关量输入通道 4

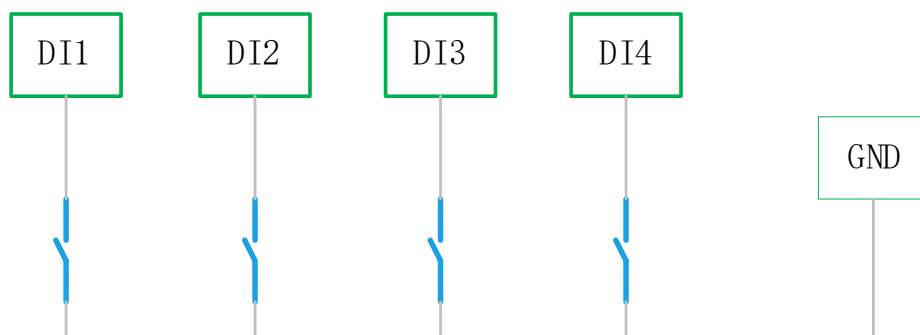
下端：

K1A： 继电器 1 常闭端  
K2A： 继电器 2 常闭端  
K3A： 继电器 3 常闭端  
K4A： 继电器 4 常闭端

K1B： 继电器 1 公共端  
K2B： 继电器 2 公共端  
K3B： 继电器 3 公共端  
K4B： 继电器 4 公共端

K1C： 继电器 1 常开端  
K2C： 继电器 2 常开端  
K3C： 继电器 3 常开端  
K4C： 继电器 4 常开端

干节点接线法：



### 3.2 LED 指示灯



上端指示灯：

POWER： 电源指示灯，默认常亮

SYS： 系统指示灯，默认 1s 闪烁一次

下端 4 个指示灯

DI1： 开关量输入通道 1，高电平输入时常亮

DI2： 开关量输入通道 2，高电平输入时常亮

DI3： 开关量输入通道 3，高电平输入时常亮

DI4： 开关量输入通道 4，高电平输入时常亮

四、 软件操作

设备参数配置教程，结合《用户测试文档》即可对设备进行简单测试

4.1 配置软件

参数配置软件介绍：



参数配置准备：

- (1) 用 USB-485 工具连接设备到电脑
- (2) 在串口配置框内配置串口波特率、停止位、校验位、数据位；（默认波特率 9600，数据位 8，停止位 1，校验位 None）



- (3) 选择串口配置框子项“命令集”

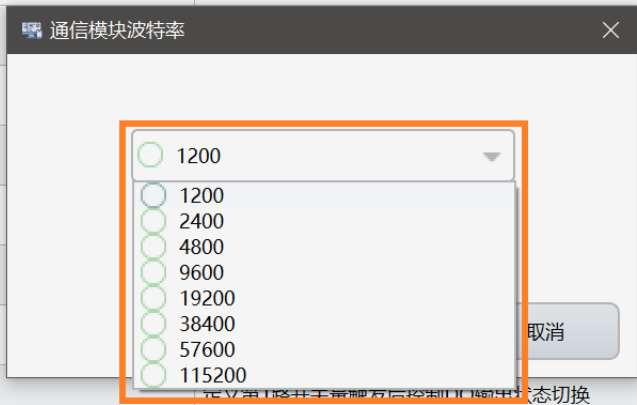


- (4) 点击“**读取参数**”命令按钮，读取设备参数（不同设备拥有不同指令集）
- (5) 双击对应参数项的“**参数值**”栏，即可对参数进行修改

## 4.2 配置基本参数

- Modbus 地址：Modbus 地址参数
- 通讯模块波特率：设备 485 通讯波特率（波特率支持主流的波特率选项）

参数名称	参数值	参数说明
<基本参数>		
Modbus地址		设备的Modbus地址，1~255
通信模块波特率	双击配置参数	与通信模块的波特率一致，一般设置为9600
<DI开关量相关参数>		
DI1告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI1告警控制		
DI2告警触发		
DI2告警控制		
DI3告警触发		
DI3告警控制		
DI4告警触发		
DI4告警控制		



## 4.3 DI 开关量相关参数

- DI1 告警触发：双击“**参数值**”栏，选择对应选项修改 DI 触发状态
- DI2 告警触发：双击“**参数值**”栏，选择对应选项修改 DI 触发状态
- DI3 告警触发：双击“**参数值**”栏，选择对应选项修改 DI 触发状态
- DI4 告警触发：双击“**参数值**”栏，选择对应选项修改 DI 触发状态

参数名称	参数值	参数说明
<基本参数>		
Modbus地址		设备的Modbus地址, 1~255
通信模块波特率		与通信模块的波特率一致, 一般设置为9600
<DI开关量相关参数>		
DI1告警触发	双击配置参数	定义第1路开关量发生何种变化后设备发出告警信息
DI1告警控制		定义第1路开关量触发后控制DO输出状态切换
DI2告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI2告警控制		定义第1路开关量触发后控制DO输出状态切换
DI3告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI3告警控制		定义第1路开关量触发后控制DO输出状态切换
DI4告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI4告警控制		定义第1路开关量触发后控制DO输出状态切换
<DI告警短信相关参数>		
安装地址		
设备身份ID		
DI1告警周期		循环告警的时间间隔, 单位为分钟, 0表示只告警一次

DI1告警触发

☐ 不告警  
☐ 不告警  
☐ 低触发  
☐ 高触发

确定 取消

- DI1 告警控制：双击“参数值”栏，选择对应选项修改 DI 告警控制
- DI2 告警控制：双击“参数值”栏，选择对应选项修改 DI 告警控制
- DI3 告警控制：双击“参数值”栏，选择对应选项修改 DI 告警控制
- DI4 告警控制：双击“参数值”栏，选择对应选项修改 DI 告警控制

<DI开关量相关参数>		
DI1告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI1告警控制	双击配置参数	定义第1路开关量触发后控制DO输出状态切换
DI2告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI2告警控制		定义第1路开关量触发后控制DO输出状态切换
DI3告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI3告警控制		定义第1路开关量触发后控制DO输出状态切换
DI4告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI4告警控制		定义第1路开关量触发后控制DO输出状态切换
<DI告警短信相关参数>		
安装地址		
设备身份ID		8位编码, 短信报警时有效
DI1告警周期		循环告警的时间间隔, 单位为分钟, 0表示只告警一次

DI1告警控制

☐ 不操作  
☐ 不操作  
☐ DO1  
☐ DO2  
☐ DO3  
☐ DO4

确定 取消

当 DIx 输入通道告警触发时，DIx 告警控制对应 DOx 继电器状态：常闭端断开，常开端闭合。

4.4 短信告警内容参数配置（设备需包含有短信模块）

注：需要配合本公司远程模块

- 安装地址： 主要用于报警通知使用（设备需包含有短信模块）
- 设备身份 ID： 主要用于报警通知使用（设备需包含有短信模块）
- DI1 告警短信内容： DI 告警内容
- DI1 恢复短信内容： DI 告警内容
- DI2 告警短信内容： DI 告警内容
- DI2 恢复短信内容： DI 告警内容
- DI3 告警短信内容： DI 告警内容
- DI3 恢复短信内容： DI 告警内容
- DI4 告警短信内容： DI 告警内容
- DI4 恢复短信内容： DI 告警内容

<DI告警短信相关参数>		
安装地址		最长16个汉字,32个字符，短信报警用
设备身份ID		8位编码，短信报警时有效
DI1告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
DI1告警短信内容		最多30个汉字或60个字符
DI1恢复短信内容		最多30个汉字或60个字符
DI2告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
DI2告警短信内容		最多30个汉字或60个字符
DI2恢复短信内容		最多30个汉字或60个字符
DI3告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
DI3告警短信内容		最多30个汉字或60个字符
DI3恢复短信内容		最多30个汉字或60个字符
DI4告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
DI4告警短信内容		最多30个汉字或60个字符
DI4恢复短信内容		最多30个汉字或60个字符

告警内容案例：

配置如下：

DI1 告警触发： 低触发

DI1 告警控制： D01

安装地址： “北厂区： ”

设备身份 ID： “05 号设备： ”

DI1 告警短信内容： “保护跳闸”

DI2 恢复短信内容： “恢复供电”

<DI开关量相关参数>		
DI1告警触发	低触发	定义第1路开关量发生何种变化后设备发出告警信息
DI1告警控制	DO1	定义第1路开关量触发后控制DO输出状态切换
DI2告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI2告警控制		定义第1路开关量触发后控制DO输出状态切换
DI3告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI3告警控制		定义第1路开关量触发后控制DO输出状态切换
DI4告警触发		定义第1路开关量发生何种变化后设备发出告警信息
DI4告警控制		定义第1路开关量触发后控制DO输出状态切换
<DI告警短信相关参数>		
安装地址	北厂区：	最长16个汉字,32个字符，短信报警用
设备身份ID	05号设备：	8位编码，短信报警时有效
DI1告警周期	1	循环告警的时间间隔，单位为分钟，0表示只告警一次
DI1告警短信内容	保护跳闸	最多30个汉字或60个字符
DI1恢复短信内容	恢复供电	最多30个汉字或60个字符
DI2告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
DI2告警短信内容		最多30个汉字或60个字符
DI2恢复短信内容		最多30个汉字或60个字符

当 DI1 检测到输入低信号时，立即触发告警控制 DO1 常闭端断开，常开端闭合，短信模块发送告警短信“北厂区：05 号设备：保护跳闸”。

当 DI1 检测到输入高信号时，等待 1s 取消告警控制 DO1 常闭端闭合，常开端断开，短信模块发送告警短信“北厂区：05 号设备：恢复供电”。

其他告警短信内容格式与低于下限值格式一致（消息内容为空项则不送内容）

五、ModbusRTU 通讯协议、组态软件软件说明

5.1 通讯协议

本产品支持标准 Modbus RTU 从站协议，能够支持标准 Modbus RTU 组态软件，详细介绍参考本公司《dtu\_modbus 协议手册》v1.01 版

5.2 寄存器地址

寄存器地址	名称	字节数	说明	备注
数字量输入				
0x0010 (16)	DI1	2	开关量 DI 输入通道 1	0000 表示无输入 0001 表示有输入
0x0011 (17)	DI2	2	开关量 DI 输入通道 2	
0x0012 (18)	DI3	2	开关量 DI 输入通道 3	
0x0013 (19)	DI4	2	开关量 DI 输入通道 4	
数字量输出				
0x00014 (20)	D01	2	数字量 D0 输出通道 1	0000 表示断开 0001 表示闭合
0x00015 (21)	D02	2	数字量 D0 输出通道 2	
0x00016 (22)	D03	2	数字量 D0 输出通道 3	
0x00017 (23)	D04	2	数字量 D0 输出通道 4	

5.3 Modbus RTU 功能码

功能码	操作	说明
01	读取单位 D0 状态	Bit 位表示 D0 输出状态
03	读取 DI，D0 寄存器值	读取 DI，D0 寄存器值
04	读取 DI，D0 寄存器值	读取 DI，D0 寄存器值
05	写单个 D0	0xFF00： 闭合;0x0000： 断开
06	写单个 D0	0x0001： 闭合;0x0000： 断开
0F	写多个 D0	参照《rtu_modbu 协议手册》v1.01
10	写多个 D0	参照《rtu_modbu 协议手册》v1.01

详细讲解参照本公司《dtu\_modbus 协议手册》v1.01

5.4 Modbus 通讯实例

(1) 读取 DO1:

a. 用 01 功能码读取 DO1

发送: 01 01 00 14 00 01 BD CE

接受: 01 01 01 00 51 88

b. 用 03 功能码读取 DO1

发送: 01 03 00 14 00 01 C4 0E

接受: 01 03 02 00 00 B8 44

c. 用 04 功能码读取 DO1

发送: 01 04 00 14 00 01 71 CE

接受: 01 04 02 00 00 B9 30

(2) 操作 DO1:

a. 用 05 功能码操作单个 DO1

发送: 01 05 00 14 FF 00 CC 3E

接受: 01 05 00 14 FF 00 CC 3E

b. 用 06 功能码操作单个 DO1

发送: 01 06 00 14 00 01 08 0E

接受: 01 06 00 14 00 01 08 0E

c. 用 0F 功能码操作多个 DO1、DO2

发送: 01 0F 00 14 00 02 01 03 AE 95

接受: 01 0F 00 14 00 02 94 0E

d. 用 10 功能码操作多个 DO1、DO2

发送: 01 10 00 14 00 02 04 00 01 00 01 63 50

接受: 01 10 00 14 00 02 01 CC

(3) 读取 DI1:

DI1 配置“低触发”给定 DI1 “低信号”

a. 用 03 功能码读取 DI1

发送: 01 03 00 10 00 01 85 CF

接受: 01 03 02 00 00 B8 44

b. 用 04 功能码读取 DI1

DI1 配置“低触发”给定 DI1 “低信号”

发送: 01 04 00 10 00 01 30 0F

接受: 01 04 02 00 00 B9 30

(4) 读取所有 DO、DI

DO1 有输出, 给定 DI2 输入信号

发送: 01 03 00 10 00 08 45 C9

接受: 01 03 10 00 00 00 01 00 00 00 00 00 01 00 00 00 00 00 00 00 F6 18

六、协议详解

地址域	功能码	数据	差错检验
-----	-----	----	------

Modbus 使用“big-Endian”表示地址和数据项，这就意味着当发射多个字节时，首先发送最高字节。

例如：寄存器地址为 0x0014，首先发送的是 0x00，然后才是 0x14。

一个正常的 Modbus 响应：响应功能码=请求功能码。

一个 Modbus 的异常响应：响应功能码=请求功能码+0x80，提供一个异常码来指示差错原因。

6.1 功能码描述

6.1.1 01 读线圈

可以使用此功能码读取继电器 D01~D04 的状态。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-4 为 0-3。

响应 PDU 中 N 个字节的线圈状态的每一个 bit 位代表一个线圈的状态，状态 1=ON，0=OFF。第一个字节的最低位 LSB 代表第 0 号线圈的状态（即起始地址指定的线圈号为 0 号线圈），其他线圈依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各线圈状态。

如果线圈数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x01
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

注：线圈状态的字节数 N=线圈数量 n/8，如果余数不等于 0，则 N=n/8+1

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x81 （请求功能码+0x80）
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量 D01-D04 的实例

请求		响应	
地址	64	地址	64
功能码	01	功能码	01
起始地址高 H	00	字节数	01
起始地址低 L	14	DO1-DO4 状态	0D

线圈数量高 H	00	CRC 校验高 H	8E
线圈数量低 L	04	CRC 校验低 L	81
CRC 校验高 H	74		
CRC 校验低 L	38		

发送：6401001400047438      DTU 响应：6401010D8E81

D01-D04 的状态字节为 0D，二进制 00001101，D01 是这个字节的 LSB(第 0 位)为 1 表示闭合，D02 是第 1 位为 0 表示断开，D03 是第 2 位为 1 表示闭合，D04 是第 3 位为 1 表示闭合，用 0 填充剩余的 4 位。

### 6.1.2 02 读离散量输入

可以使用此功能码读取数字量输入 DI1~DI4 的状态。

请求 PDU 详细说明了起始地址，即指定第一个输入量的地址和数量，从零开始寻址，因此寻址输入量 1-4 为 0-3。

响应 PDU 中 N 个字节的输入量状态的每一个 bit 位代表一个输入量的状态，状态 1=HIGH，0=LOW。第一个字节的最低位 LSB 代表第 0 号输入量的状态（即起始地址指定的输入量为 0 号输入量），其他输入量依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各输入量状态。

如果输入量数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x02
起始地址	2 个字节	0x0010 至 0x0013
输入数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x02
字节数	1 个字节	N
输入状态	N 个字节	$N=n/8$ ，或 $N=n/8+1$
CRC 校验	2 个字节	

注：输入状态的字节数  $N=$ 输入数量  $n/8$ ，如果余数不等于 0，则  $N=n/8+1$

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x82 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量输入 DI1-DI4 的实例

请求		响应	
地址	64	地址	64
功能码	02	功能码	02
起始地址高 H	00	字节数	01
起始地址低 L	10	DI1-DI4 状态	0C
输入数量高 H	00	CRC 校验高 H	BF
输入数量低 L	04	CRC 校验低 L	41
CRC 校验高 H	71		
CRC 校验低 L	F9		

发送：64020010000471F9

DTU 响应：6402010CBF41

DI1-DI4 的状态字节为 0C，二进制 00001100，DI1 是这个字节的 LSB(第 0 位)为 0 表示低，DI2 是第 1 位为 0 表示低，DI3 是第 2 位为 1 表示高，DI4 是第 3 位为 1 表示高，用 0 填充剩余的 4 位。

### 6.1.3 03 读保持寄存器

### 6.1.4 04 读输入寄存器

使用该功能码可以读取所有寄存器包括 DI1-DI4、DO1-DO4 的状态。

请求 PDU 详细说明了起始寄存器地址和寄存器数量，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

响应报文中的寄存器数据每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2。对于 DI1-DI4，2 个字节的值 0000 代表输入量为低，0001 代表输入量为高，对于 DO1-DO4，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 04
起始地址	2 个字节	0x0010 至 0x0017
寄存器数量	2 个字节	n(1 至 8)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 0x04
字节数	1 个字节	$N=2*n$
寄存器值	N 个字节	$N=2*n$ , n 为寄存器数量
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
----	-------	--

功能码	1 个字节	0x83 或 0x84 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读模拟量输入 AI1-AI8、离散量输入 DI1-DI4、继电器状态 DO1-DO4 的

请求		响应	
地址	64	地址	64
功能码	03	功能码	03
起始地址高 H	00	字节数	10
起始地址低 L	10	DI1-DI4 状态	8 个字节
寄存器数量高 H	00	DO1-DO4 状态	8 个字节
寄存器数量低 L	08	CRC 校验高 H	
CRC 校验高 H	4C	CRC 校验低 L	
CRC 校验低 L	35		

发送: 6403001000084C3C

### 6.1.5 05 写单个线圈

可以使用该功能码写单个继电器 DO1-DO4 为断开或闭合

请求数据域中的常量说明请求的 ON/OFF 状态, 十六进制值 0xFF00 请求输出为 ON(闭合), 十六进制值 0x0000 请求输出为 OFF(断开), 其他所有值都是非法的, 对输出不起作用, DTU 返回错误响应。

请求域中的输出地址规定了要写入线圈的地址。

正常响应是请求的应答, 在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0017
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0017
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x85 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 D02 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	05	功能码	05
输出地址高 H	00	输出地址高 H	00
输出地址低 L	15	输出地址低 L	15
输出值高 H	FF	输出值高 H	FF
输出值低 L	00	输出值低 L	00
CRC 校验高 H	94	CRC 校验高 H	94
CRC 校验低 L	0B	CRC 校验低 L	0B

发送：64050015FF00940B      DTU 响应：64050015FF00940B

### 6.1.6 06 写单个寄存器

可以使用该功能码写单个继电器 D01-D04 为断开或闭合。

请求数据域中的寄存器值说明请求的 ON/OFF 状态，十六进制值 0001 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)。

请求域中的寄存器地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0017
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0017
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x86 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 D02 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64

功能码	06	功能码	06
寄存器地址高 H	00	寄存器地址高 H	00
寄存器地址低 L	15	寄存器地址低 L	15
寄存器值高 H	00	寄存器值高 H	00
寄存器值低 L	01	寄存器值低 L	01
CRC 校验高 H	50	CRC 校验高 H	50
CRC 校验低 L	3B	CRC 校验低 L	3B

发送：640600150001503B

DTU 响应：640600150001503B

#### 6.1.7 0F 写多个线圈

可以使用此功能码写多个继电器 D01~D04 为断开或闭合。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-4 为 0-3。

请求数据域中的内容说明了被请求的 ON/OFF 状态，域比特位中的逻辑“1”请求相应输出为 ON，域比特位中的逻辑“0”请求相应输出为 OFF。从数据域中第一个字节的 bit0 开始到 bit7，然后到第二个字节的 bit0，依次表示第一个线圈到第 n 个线圈的 ON/OFF 值。

正常响应返回功能码、起始地址和线圈数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
字节数	1 个字节	$N=n/8$ , 或 $N=n/8+1$
输出值	N 个字节	
CRC 校验	2 个字节	

注：线圈输出字节数  $N=$ 线圈数量  $n/8$ ，如果余数不等于 0，则  $N=n/8+1$

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x8F (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求从线圈 D01 开始写入 4 个线圈的实例

请求		响应	
地址	64	地址	64
功能码	0F	功能码	0F
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
线圈数量高 H	00	线圈数量高 H	00
线圈数量低 L	04	线圈数量低 L	04
字节数	01	CRC 校验高 H	1D
输出值	0A	CRC 校验低 L	F9
CRC 校验高 H	48		
CRC 校验低 L	85		

发送：640F00140004010A4885

DTU 响应：640F001400041DF9

D01-D04 的输出值为 0A，二进制 00001010，D01 是这个字节的 LSB(第 0 位)为 0 表示断开，D02 是第 1 位为 1 表示闭合，D03 是第 2 位为 0 表示断开，D04 是第 3 位为 1 表示闭合，用 0 填充剩余未使用的 4 位。

#### 6.1.8 10 写多个寄存器

使用该功能码可以写连续寄存器 D01-D04 的状态。

请求 PDU 详细说明了起始寄存器地址、寄存器数量、字节数和寄存器值，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

寄存器数据中每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0017
寄存器数量	2 个字节	n(1 至 4)
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n, n 为寄存器数量
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0017
寄存器数量	2 个字节	n(1 至 4)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x90 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个控制继电器 D01-D04 的实例

请求		响应	
地址	64	地址	64
功能码	10	功能码	10
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
寄存器数量高 H	00	寄存器数量高 H	00
寄存器数量低 L	04	寄存器数量低 L	04
字节数	08	CRC 校验高 H	88
DO1 寄存器值高 H	00	CRC 校验低 L	3B
DO1 寄存器值高 L	01		
DO2 寄存器值高 H	00		
DO2 寄存器值高 L	00		
DO3 寄存器值高 H	00		
DO3 寄存器值高 L	00		
DO4 寄存器值高 H	00		
DO4 寄存器值高 L	01		
CRC 校验高 H	F2		
CRC 校验低 L	61		

发送：6410001400040800010000000000001      DTU 响应：641000140004883B

D01 寄存器值为 0001 表示闭合，D02 寄存器值为 0000 表示断开，D03 寄存器值为 0000 表示断开，D04 寄存器值为 0001 表示闭合。

## 6.2 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，众山 DTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码 04：表示读写寄存器错误。

## 6.3 CRC 校验算法

CRC 即[循环冗余校验码](#) (Cyclic Redundancy Check)：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

附 CRC 校验算法代码：

```
uint16_t mb_crc( uint8_t *snd, uint16_t num )
{
    uint8_t CRC_Lb, CRC_Hb;
    uint8_t CRC_L, CRC_H;
    uint16_t crc;

    CRC_H = 0xFF;
    CRC_L = 0xFF;

    for ( uint16_t i = 0; i < num; i++ ) {
        CRC_L = CRC_L ^ snd[i];
        for ( uint16_t j = 0; j < 8; j++ ) {
            CRC_Lb = CRC_L;
            if ((CRC_L & 1) == 1) {
                CRC_L = (CRC_L - 1) / 2;
                CRC_Lb = CRC_L;
                CRC_Hb = CRC_H;
                if ( (CRC_H & 1) == 1 ) {
                    CRC_L = CRC_L + 128;
                    CRC_Lb = CRC_L;
                    CRC_H = (CRC_H - 1) / 2;
                    CRC_Hb = CRC_H;
                } else {
                    CRC_H = CRC_H / 2;
                    CRC_Hb = CRC_H;
                }
            }
            CRC_L = CRC_L ^ 1;
            CRC_Lb = CRC_L;
            CRC_H = CRC_H ^ 0xA0;
            CRC_Hb = CRC_H;
        }
        else
        {
            CRC_L = CRC_L / 2;
            CRC_Lb = CRC_L;
```

```
CRC_Hb = CRC_H;
if ( (CRC_H & 1) == 1 ) {
    CRC_L = CRC_L + 128;
    CRC_Lb = CRC_L;
    CRC_H = (CRC_H - 1) / 2;
    CRC_Hb = CRC_H;
} else {
    CRC_H = CRC_H / 2;
    CRC_Hb = CRC_H;
}
}
}
}
crc = CRC_L;
crc <<= 8;
crc |= CRC_H;
return crc;
}
```

## 七、附录

V1.3

- \* 更新 modbus 协议详解

v1.2

- \* 更新接线图新增干接点接线法，修改产品功能、特性描述

v1.1

- \* 插接线图，外观尺寸，指示灯

v1.0

- \* 编写 4 路开关量说明书